eip ESCUELA INTERNACIONAL POSGRADOS

Máster en Programación avanzada en Python para Hacking, BigData y Machine Learning

Fundamentos de IA y Machine Learning

LECCIÓN 05

Aprendizaje no supervisado - *Clustering*

ÍNDICE

INTRODUCCIÓN

Como se comentó previamente, el aprendizaje no supervisado es aquel empleado en bases de datos no etiquetadas. En esta lección, nos centraremos en los algoritmos de agrupamiento.

OBJETIVOS

Al finalizar esta lección serás capaz de:

1. Saber los fundamentos del aprendizaje no supervisado y, en concreto, del *clustering*.
2. Conocer las características fundamentales del *clustering* particional, jerárquico y basado en densidades.
3. Aplicar los distintos tipos de agrupamiento e interpretar los resultados.

1. Introducción – Aprendizaje no supervisado

Los métodos de aprendizaje no supervisado no necesitan datos etiquetados, sino que tratan de descubrir la estructura de estos. Puede ser un objetivo por sí solo o un medio para un fin. Existen distintos objetivos enmarcados dentro del aprendizaje no supervisado:

- **Agrupación**: dados dos ejemplos sin etiquetar, agruparlos siguiendo algún criterio predefinido.

- **Generación de jerarquías**: dados unos datos en un mismo nivel, generar las jerarquías que organicen dichos datos.

- **Reducción de dimensionalidad**: dados unos datos, reducir la dimensión o número de atributos que caracterizan dichos datos.

- **Visualización**: dados unos datos con representación compleja, permitir su visualización.

1. Introducción – Agrupamiento

Objetivo:

- Agrupar instancias o patrones creando *clusters* similares.
- Es decir, segmentar una población heterogénea en un número de subgrupos homogéneos o *clusters.*

Cluster: grupo o conjunto de objetos o patrones.

- Similares entre sí (a los de su *cluster*).
- Distintos a los de otro *cluster.*

	Features			
Objects	F1	F2	F3	...
obj1	.	.	.	...
obj2	.	.	.	...
⋮	⋮	⋮	⋮	⋱

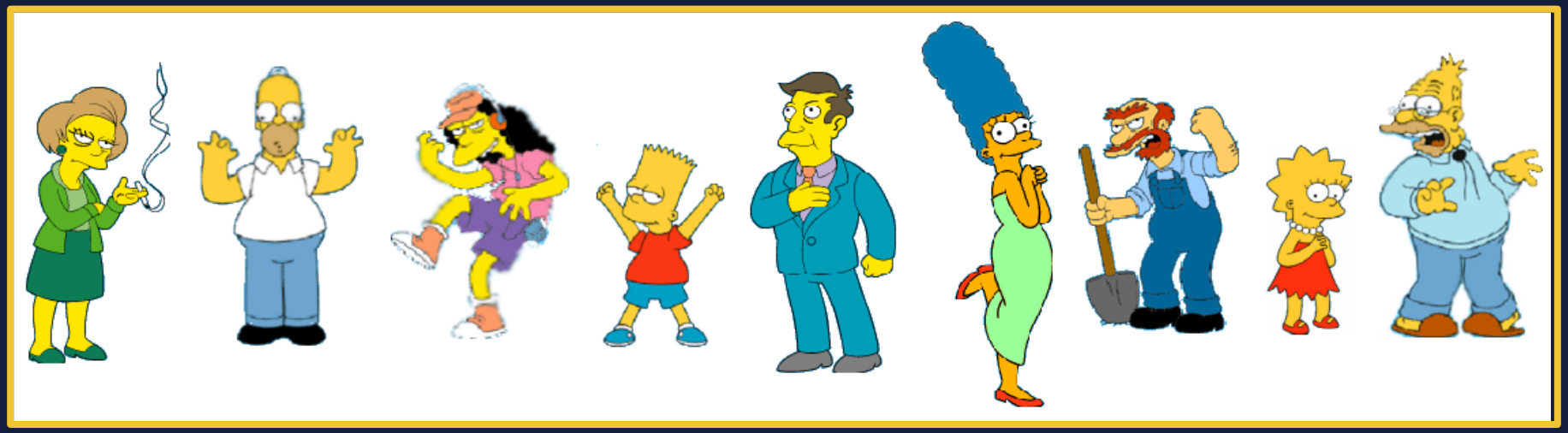
1. Introducción – *Clustering vs* Clasificación

- Ambos tratan de asignar la clase o *cluster* apropiado a un determinado ejemplo.

- Sin embargo, en *clustering* no existen clases predefinidas, por lo que el objetivo es agrupar los datos, en lugar de desarrollar clasificadores (no se realizarán particiones de los datos y será más complicada la evaluación de una determinada metodología).

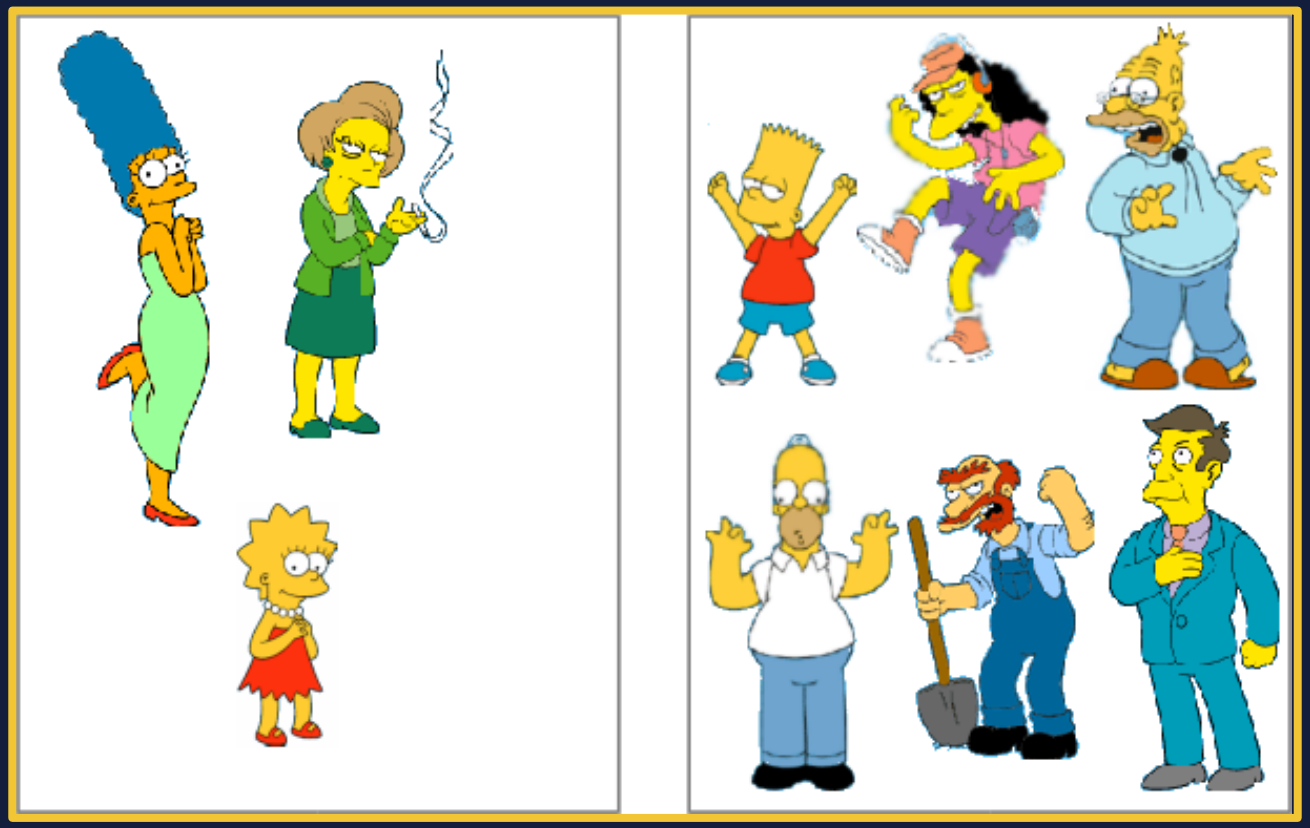
1. Introducción – Ejemplo

¿Cuál sería la forma natural de agrupar a los personajes de los *Simpsons*?



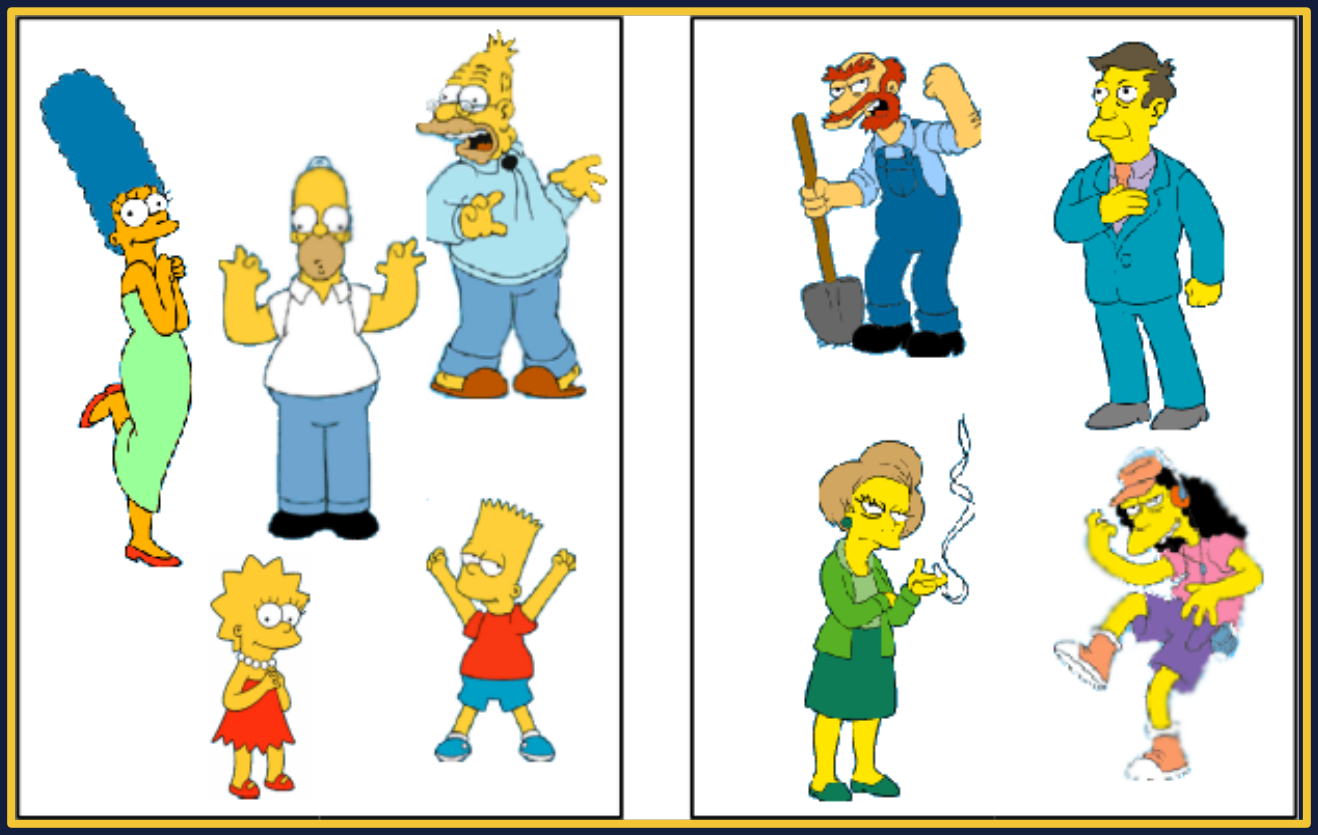
1. Introducción – Ejemplo

Hombres vs Mujeres



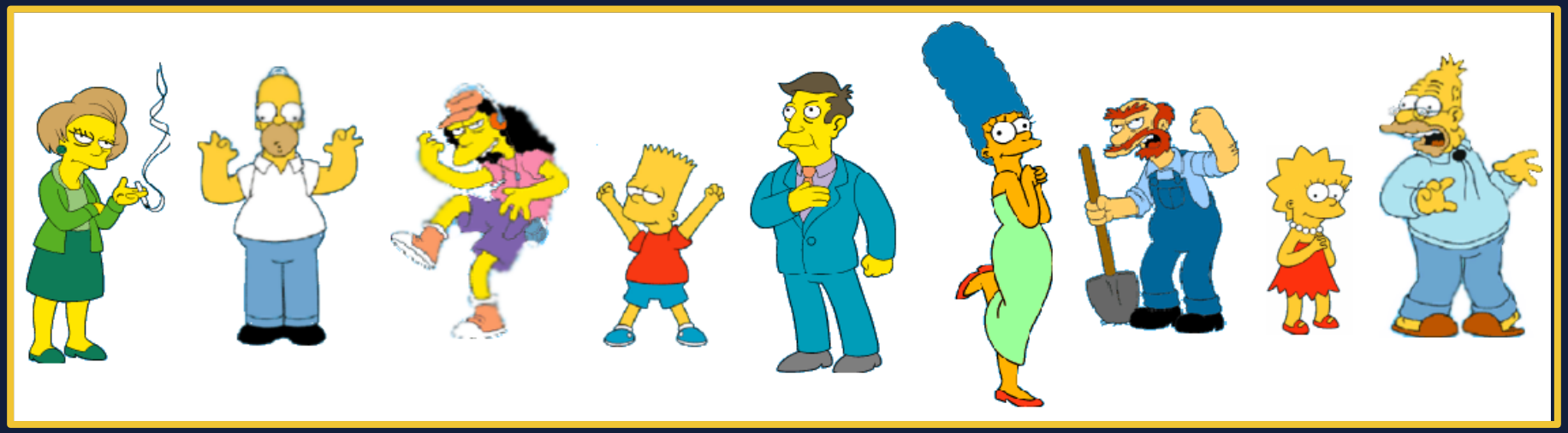
1. Introducción – Ejemplo

Familia Simpsons vs Trabajadores de la escuela



1. Introducción – Ejemplo

- Depende de con qué objetivo, una opción es mejor que otra.
- El *clustering* es subjetivo.



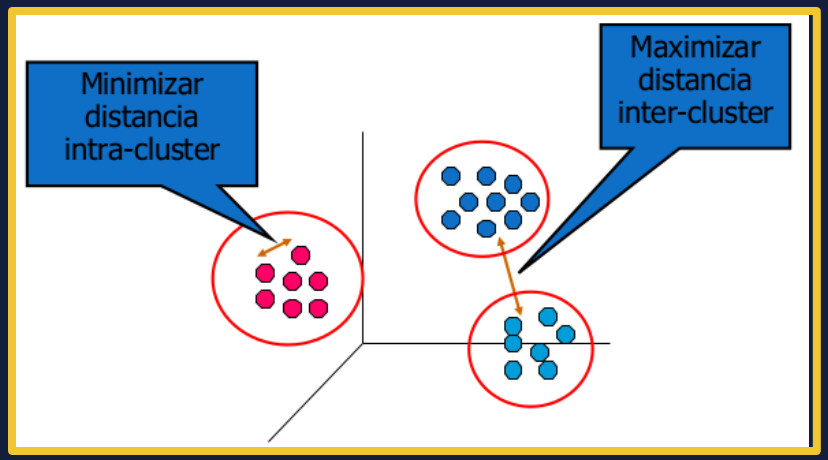
1. Introducción – Aplicaciones del agrupamiento

Existen distintos campos donde se aplica el agrupamiento:

- **Marketing**: identificar grupos de clientes con comportamiento similar (segmentación de mercado) o posicionamiento de productos.
- **Internet**: agrupamiento de textos, vídeos, imágenes…
- **Genética**: agrupamiento de genes para inferir estructuras en la población.
- **Redes sociales**: identificar comunidades.
- **Sistemas de recomendación**: recomendar productos similares a los gustos personales.
- **Ciencias sociales**: minería de datos educativos, tipologías de opiniones, etc.

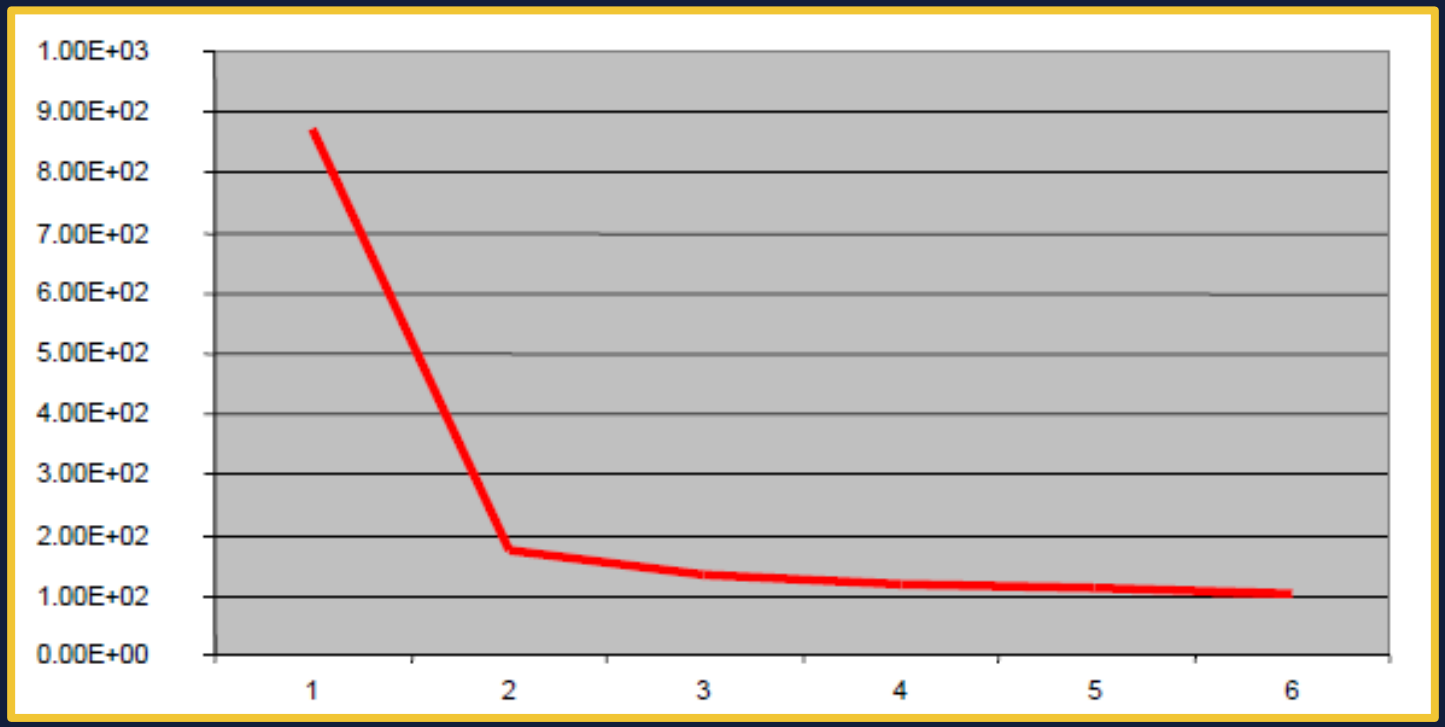
1. Introducción – Medidas de similitud

- Debe haber más cercanía entre las distancias dentro del *cluster* que respecto a las que están fuera del mismo.
- Hablamos de similitud entre instancias, pero ¿qué es la similitud y cómo la medimos? Usualmente, se expresa en términos de distancias, dónde $d(i,j) > d(i,k)$ nos indica que el objeto i es más parecido a k que a j.
- La definición de la métrica de similitud/distancia será distinta en función del tipo de dato y de la interpretación semántica que nosotros hagamos. En otras palabras, la similitud entre objetos es subjetiva.



1. Introducción – Evaluación de los métodos de agrupamiento

$$SSE = \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu}_{x_i})^2$$



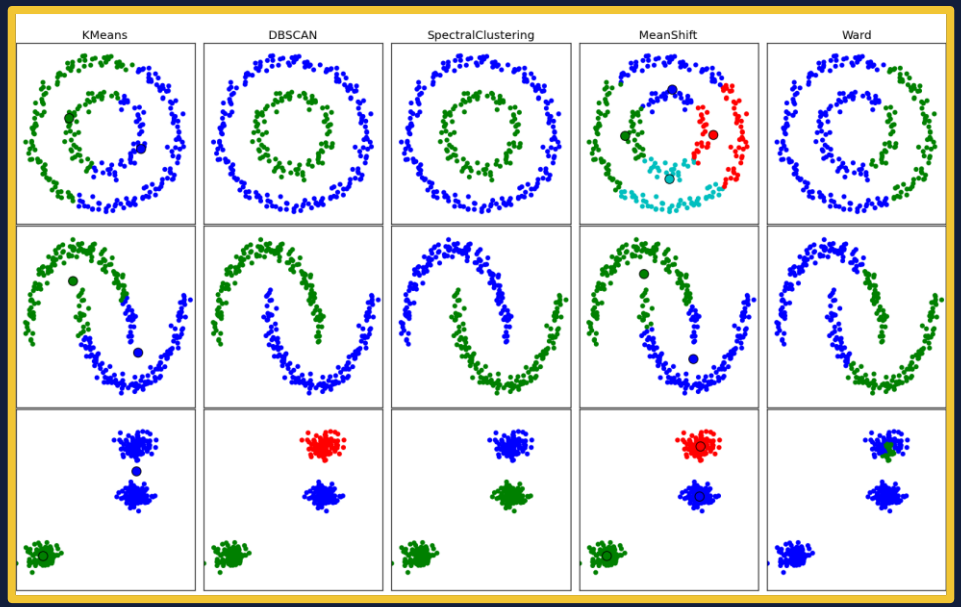
1. Introducción – Requisitos para el algoritmo "perfecto"

- Sea escalable al tamaño de los datos.
- Maneje distintos tipos de datos.
- Identifique *clusters* con formas arbitrarias.
- Tenga un número mínimo de parámetros.
- Sea tolerante al ruido y a *outliers*.
- Sea independiente al orden de presentación de los patrones de entrenamiento.
- Permita trabajar con espacios con muchas dimensiones diferentes.
- Sea interpretable.

1. Introducción – Tipos de agrupamiento

- Particionales.
- Jerárquicos.
- Otros: basados en densidades, en modelos, etc.

Distintos tipos de algoritmos llevarán a distintos *clusters* → Estudiar bien el problema.



2. Agrupamiento particional – k-medias

- Se asume que se conoce o se cree conocer el número de *clusters* k.
- Dado este valor de k, la idea es encontrar la partición que optimice el criterio de particionamiento.
- Cada grupo tiene asociado un centroide, y se busca que cada punto esté cerca de su centroide.
- Iterativamente se van actualizando los centroides y reasignado patrones hasta que dejen de cambiar.

2. Agrupamiento particional – Procedimiento general

1. Seleccionar las k semillas iniciales, es decir, los centroides iniciales: se pueden utilizar patrones existentes o puntos dentro del espacio de los patrones.

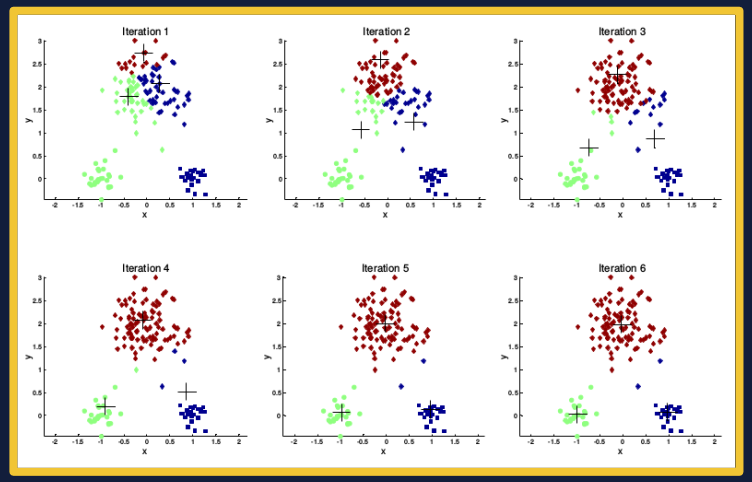
2. Asignar cada patrón al *cluster* (centroide) más cercano: usando una de las distancias vistas anteriormente.

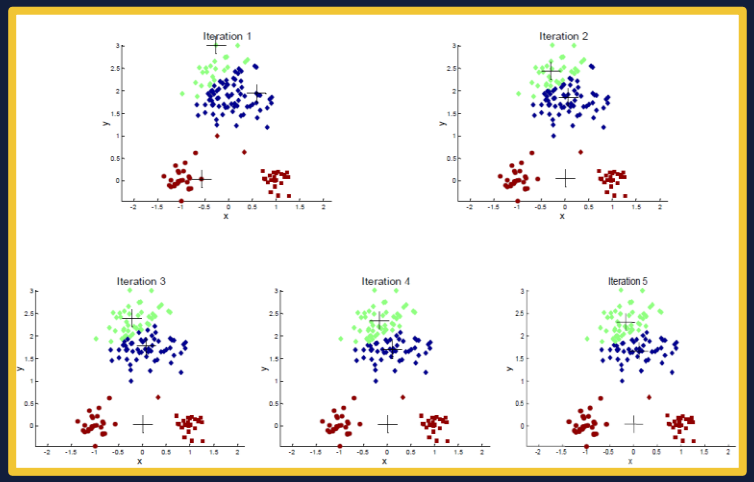
3. Actualizar los centroides: el algoritmo k-medias calcula el centroide como el punto medio de todos los patrones del *clusters*. Existen otras variantes como el algoritmo k-medianas que utiliza, como su nombre indica, la mediana en lugar de la media.

4. Repetir el proceso hasta que los centroides no cambien o la asignación de los patrones sea la misma, lo que indica que el algoritmo ha convergido

2. Agrupamiento particional – Ventajas e incovenientes

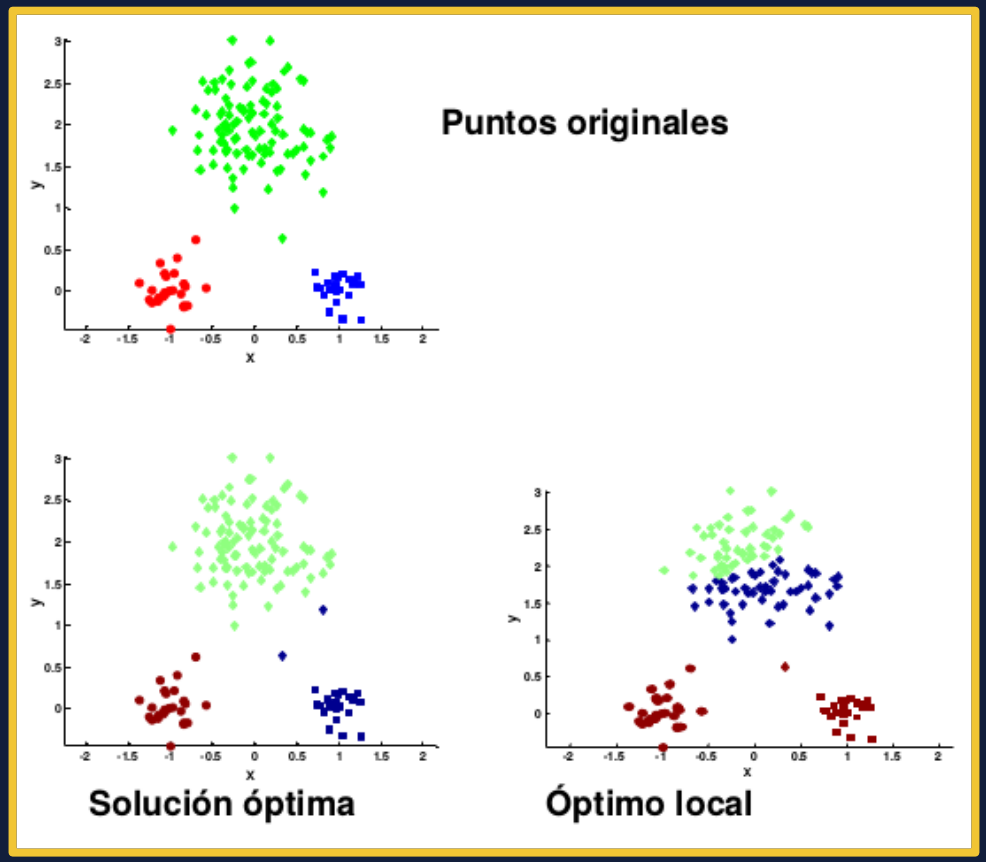
- Es eficiente.
- Se necesita saber el número de agrupamientos k.
- Incapacidad de detectar ruido u *outliers*.
- El resultado final depende de la inicialización, y además, es muy sensible a esta.





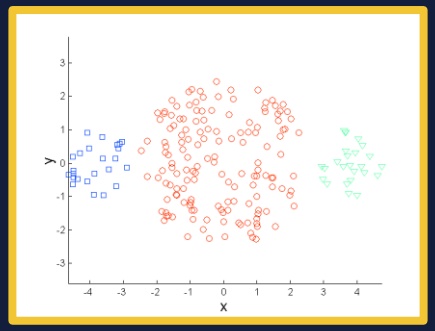
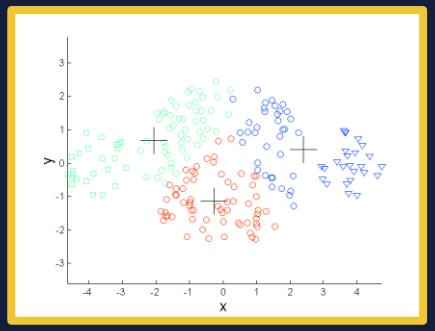
2. Agrupamiento particional – Ventajas e incovenientes

- Finaliza en un óptimo local.

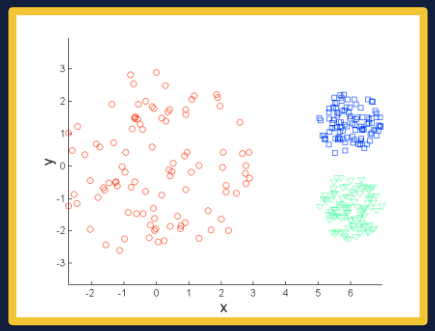
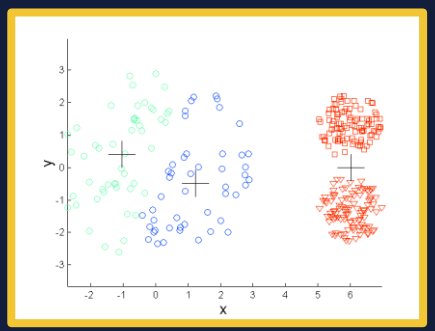


2. Agrupamiento particional – Ventajas e incovenientes

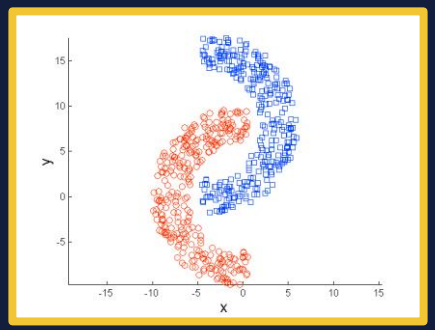
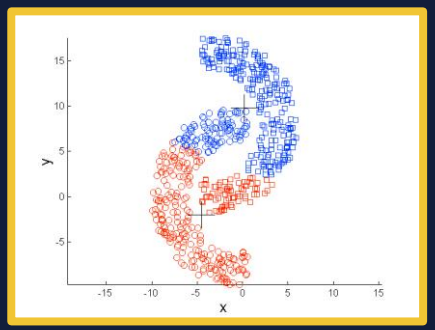
Clusters de distinto tamaño

Clusters de distinta densidad

Clusters no convexos

2. Agrupamiento particional – Ejemplo

Se pide agrupar un total de 8 patrones bidimensionales en tres *clusters* ($k = 3$). Los patrones son los siguientes: A1 (2,10), A2(2,5), A3(8,4), A4(5,8), A5(7,5), A6(6,4), A7(1,2) y A8(4,9). Los centroides iniciales son los puntos A1, A4 y A7. La métrica de distancia utilizada será la distancia euclídea.

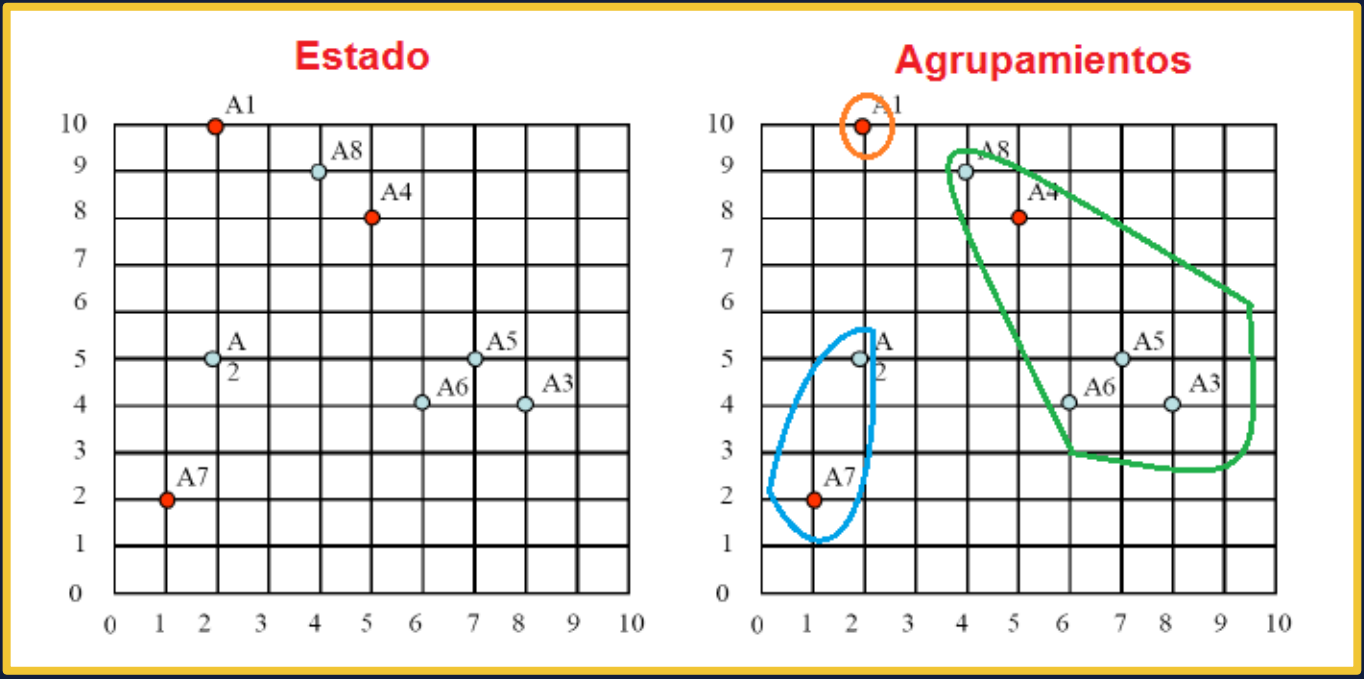
Se pide:

1. Representar los *clusters* creados y la posición de los centroides después de cada iteración.
2. El valor de la métrica *SSE*.

2. Agrupamiento particional – Ejemplo

Estado inicial

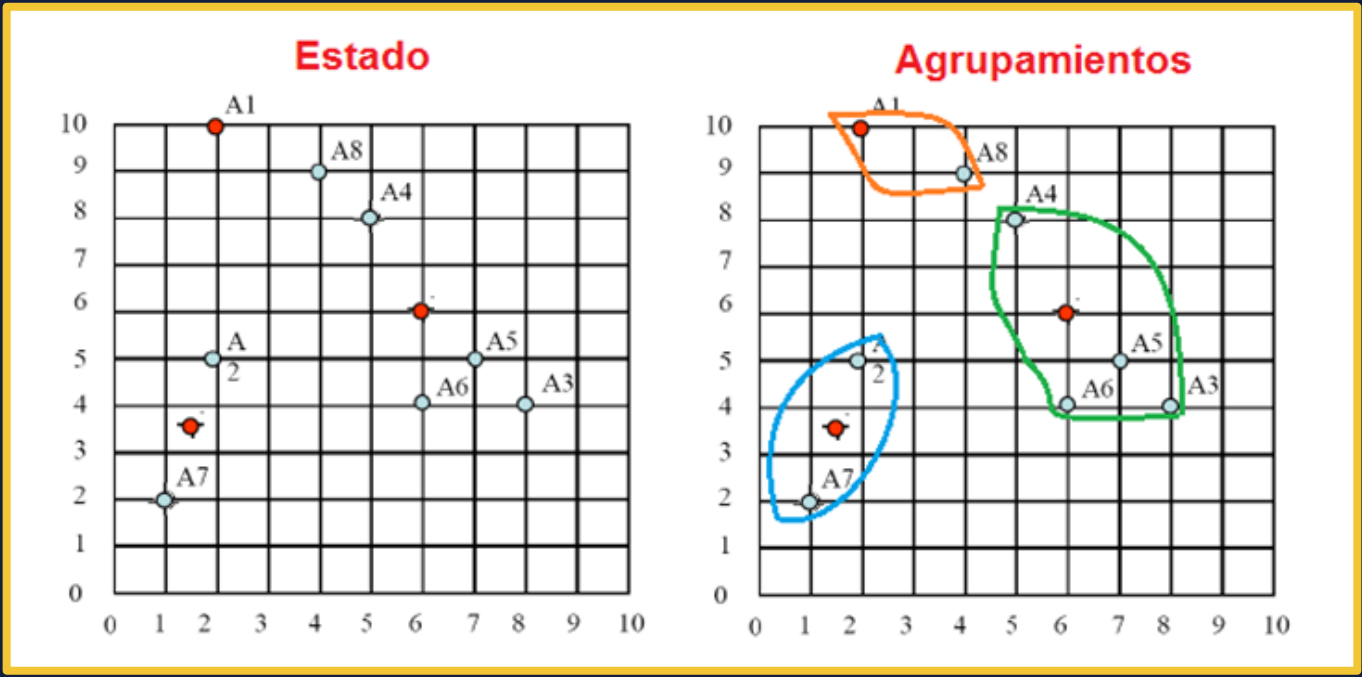
C1: (2,10) C2: (1,2) C3:(5,8)



2. Agrupamiento particional – Ejemplo

Iteración 1

C1: (2,10) C2: ((1+2) /2, (2+5) /2) = (1.5,3.5) C3: ((4+5+6+7+8) /5, (9+8+4+5+4) /5) = (6,6)

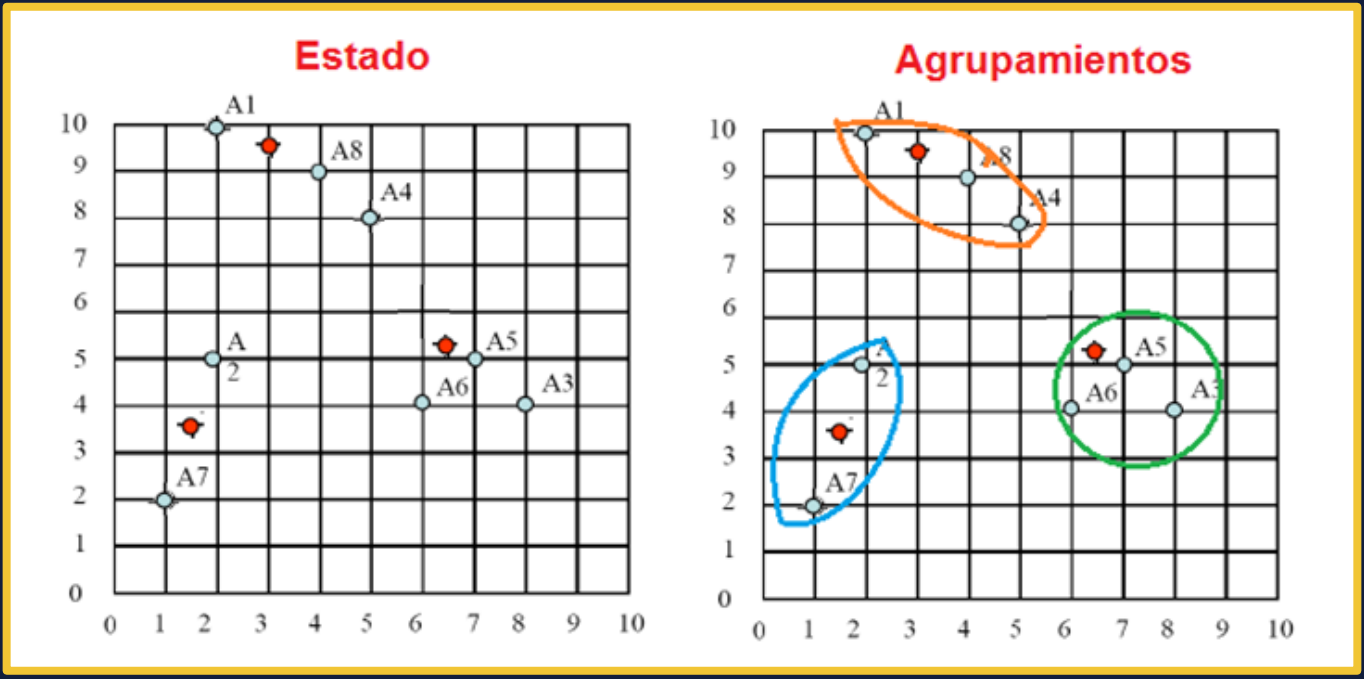


2. Agrupamiento particional – Ejemplo

Iteración 2

C1: ((2+4) /2, (10+9) /2) = (3,9.5) C2: ((1+2) /2, (2+5) /2) = (1.5,3.5)

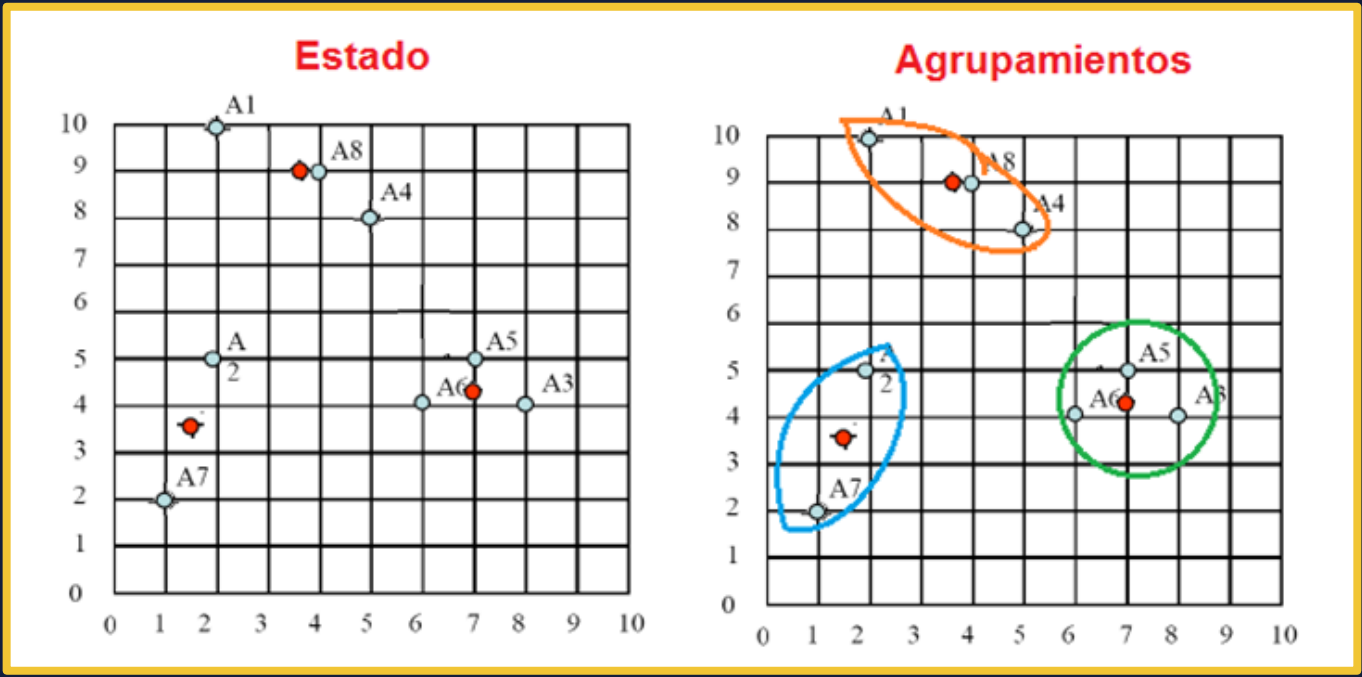
C3: ((5+6+7+8) /4, (8+4+5+4) /4) = (6.5,5.25)



2. Agrupamiento particional – Ejemplo

Iteración 2

C1: ((2+4+5) /3, (10+9+8) /3) = (3.66,9) C2: (1.5,3.5) C3: ((6+7+8) /3, (4+5+4) /3) = (7,4.33)



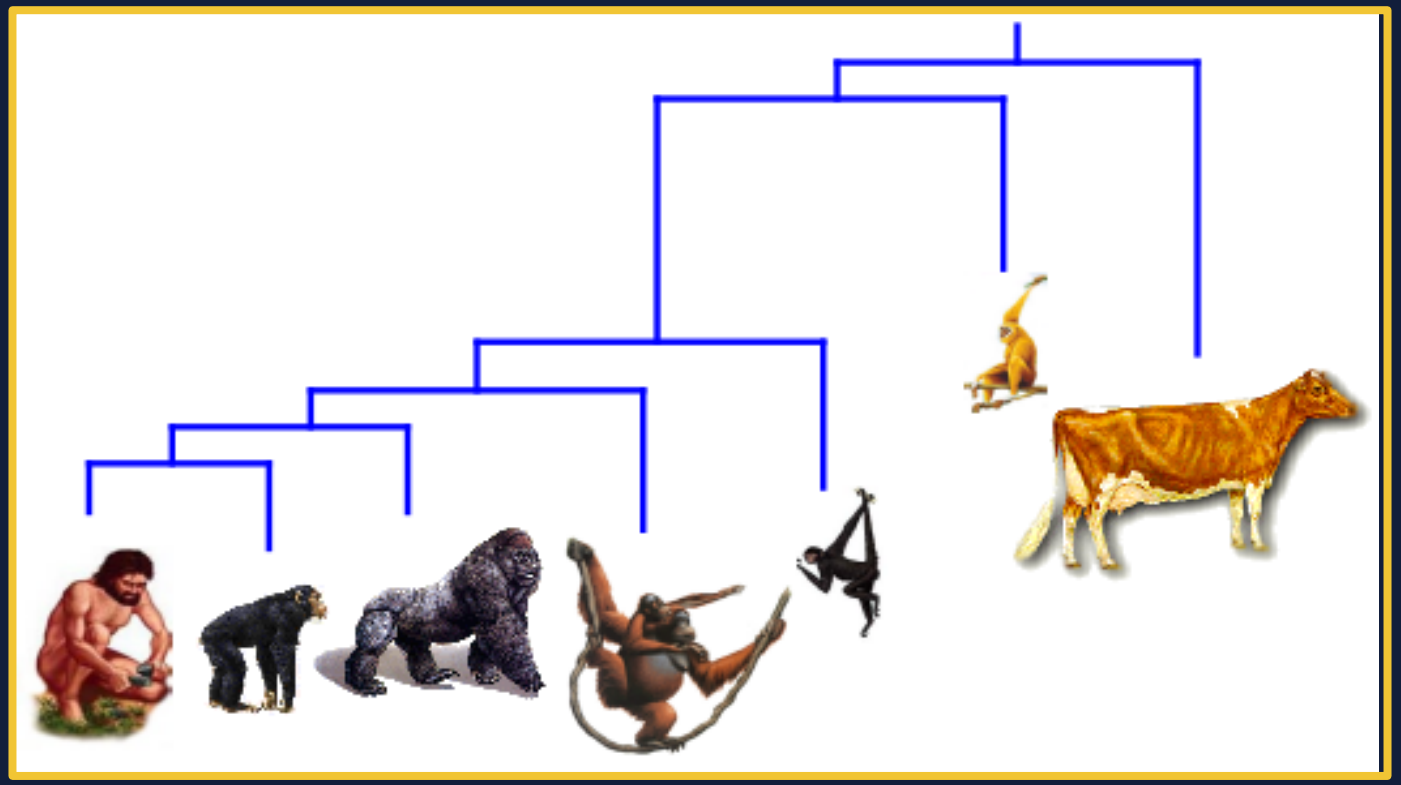
2. Agrupamiento particional – Ejemplo

Cálculo del *SSE*

Punto	*Cluster*	Distancia al centroide
A1 (2,10)	*Cluster* 1: C1(3.66,9)	1.938
A2 (2,5)	*Cluster* 2: C2(1.5,3.5)	1.5811
A3 (8,4)	*Cluster* 3: C3(7,4.33)	1.053
A4 (5,8)	*Cluster* 1: C1(3.66,9)	1.672
A5 (7,5)	*Cluster* 3: C3(7,4.33)	0.67
A6 (6,4)	*Cluster* 3: C3(7,4.33)	1.053
A7 (1,2)	*Cluster* 2: C2(1.5,3.5)	1.5811
A8 (4,9)	*Cluster* 1: C1(3.66,9)	0.34
	SSE	**14.3333**

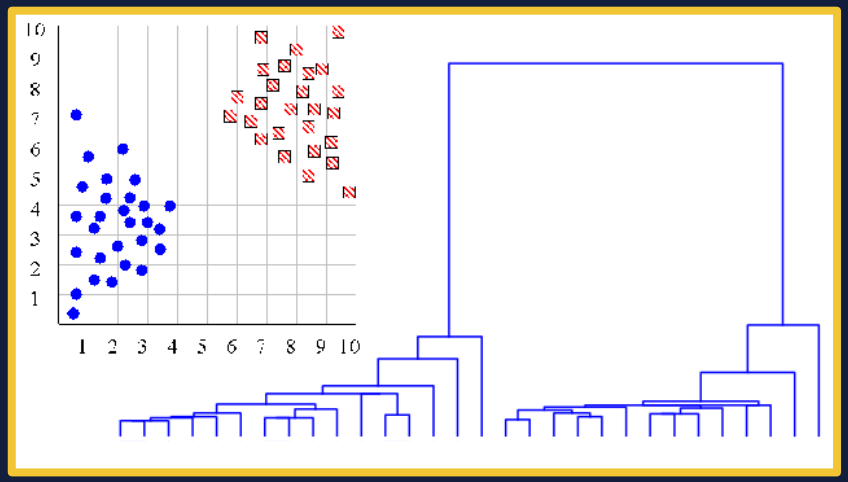
3. Agrupamiento jerárquico – Dendograma

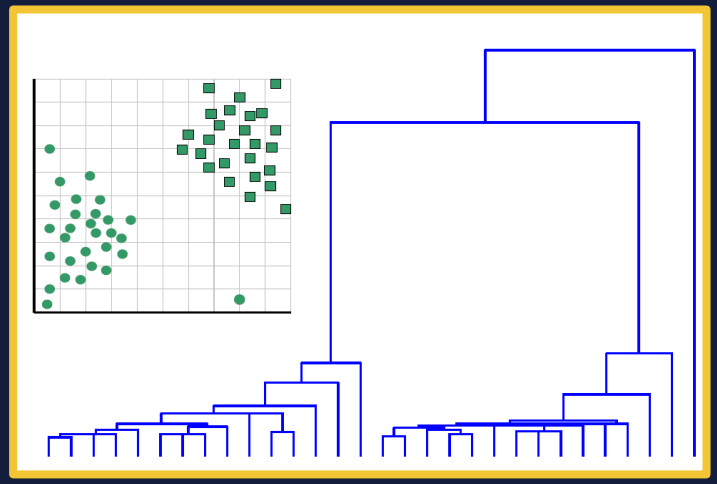
- El dendograma nos permite representar la similitud entre los objetos y el cómo se van uniendo en *clusters* por niveles.



3. Agrupamiento jerárquico – Dendograma

- Nos permite decidir el valor de k.
- Nos permite detectar *outliers.*





NO SERÁ FÁCIL

3. Agrupamiento jerárquico – Tipos

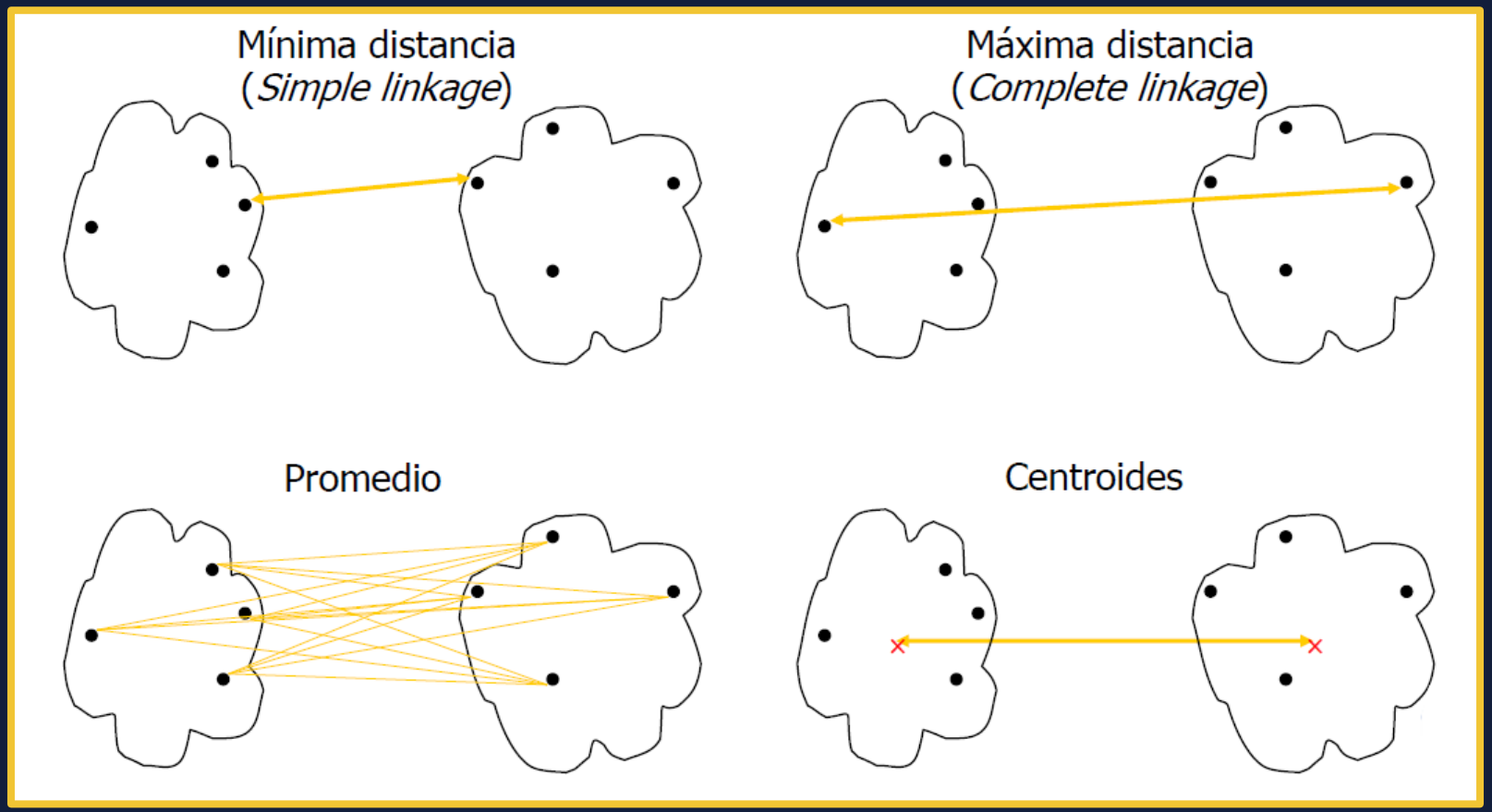
Existen dos tipos de *clustering* jerárquicos:

- **Aglomerativo**: la situación de partida suele ser un *cluster* por cada patrón. En cada paso se fusionan los dos *clusters* más cercanos. El proceso se repite hasta llegar a un único *cluster*.

- **Divisivo**: es el proceso contrario, se comienza con único *cluster* que se va dividiendo en cada paso hasta tener un *cluster* por patrón. Este proceso necesita de un mayor número de cálculos por lo que, por norma general, se aplica el aglomerativo.

Como se observa, la salida es una jerarquía entre *clusters*, donde el nivel de corte nos llevará a *clustering* distintos. De este modo, no es necesario fijar el valor de k.

3. Agrupamiento jerárquico – Distancia entre *clusters*



3. Agrupamiento jerárquico – Ventajas e inconvenientes

- Baja escalabilidad.
- No es necesario establecer el valor de k.
- Permite la detección de *outliers*.
- No depende de la inicialización.
- Permite detectar *clusters* no convexos.

3. Agrupamiento jerárquico – Ejemplo

Dada la siguiente matriz de distancias entre cuatro patrones:

	A	B	C	D
A	0	1	4	5
B		0	2	6
C			0	3
D				0

Se pide:

1. Aplicar un *clustering* jerárquico con el método de enlace simple.
2. Aplicar un *clustering* jerárquico con el método de enlace completo.

3. Agrupamiento jerárquico – Ejemplo

Enlace simple

Iteración 1

Elegimos la mínima distancia entre los *clusters:*

$d(A,B) = 1$ → es la mínima por lo que unimos esos dos puntos en el *cluster*. Tendríamos los siguientes *clusters*: $C1(A,B) - C2(C) - C3(D)$

Iteración 2

$d(C1,C2) = \min\big(d(A,C), d(B,C)\big) = \min(4,2) = 2$ → Unimos porque es la mínima.

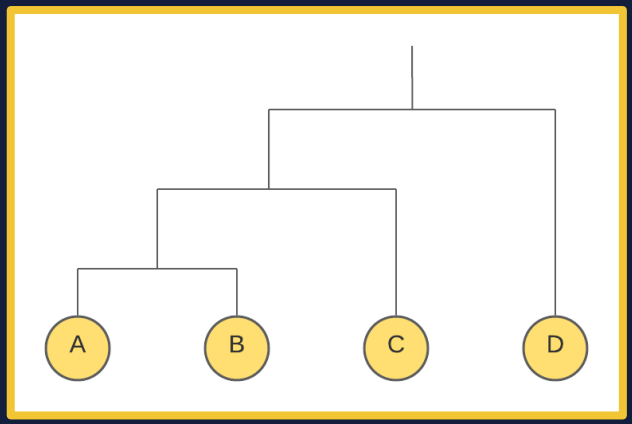
$d(C1,C3) = \min\big(d(A,D), d(B,D)\big) = \min(5,6) = 5$

$d(C2,C3) = 3$

Tendríamos los siguientes *clusters*: $C1(A,B,C) - C2(D)$

Iteración 3

Unimos los dos *clusters* que quedan: $C(A,B,C,D)$



3. Agrupamiento jerárquico – Ejemplo

Enlace completo

Iteración 1

Elegimos la mínima distancia entre los *clusters:*

$d(A,B) = 1$ → es la mínima por lo que unimos esos dos puntos en el *cluster*. Tendríamos los siguientes *clusters*: $C1(A,B) - C2(C) - C3(D)$

Iteración 2

$d(C1,C2) = \max\big(d(A,C), d(B,C)\big) = \max(4,2) = 4$

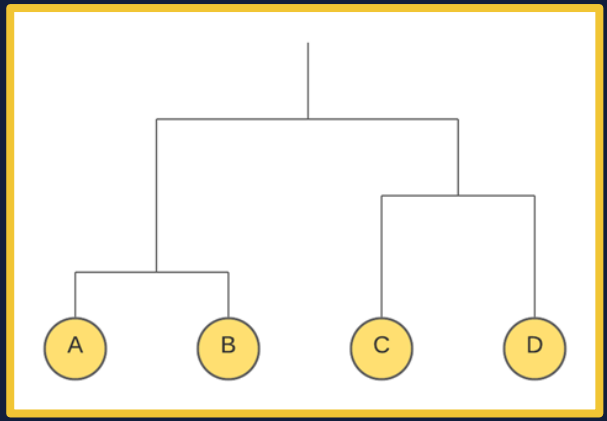
$d(C1,C3) = \max\big(d(A,D), d(B,D)\big) = \max(5,6) = 6$

$d(C2,C3) = 3$ → Unimos porque es la mínima.

Tendríamos los siguientes *clusters*: $C1(A,B) - C2(C,D)$

Iteración 3

Unimos los dos *clusters* que quedan: $C(A,B,C,D)$



4. Agrupamiento basado en densidades

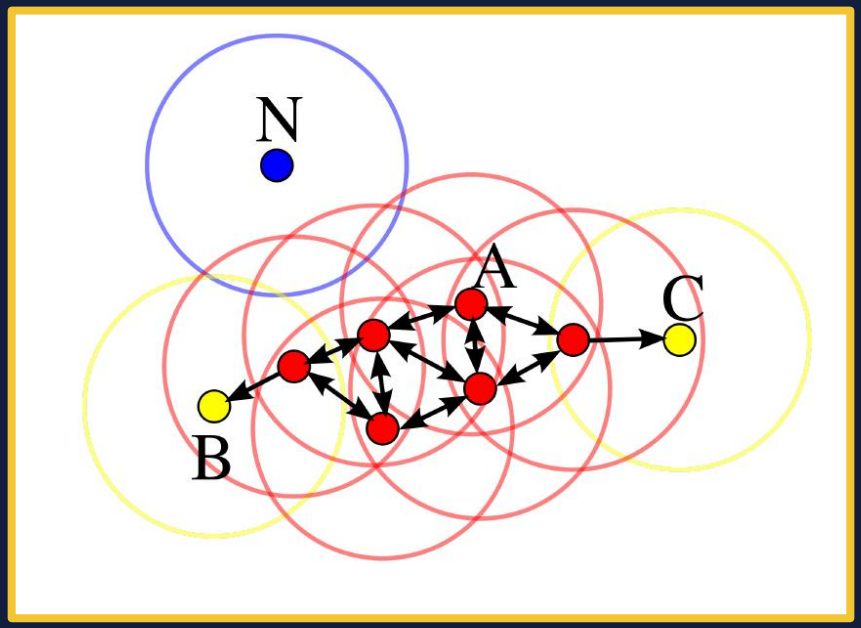
- El agrupamiento basado en densidades se fundamenta en el siguiente principio: agrupar regiones densas de puntos separadas de otras regiones densas mediante regiones poco densas.

- A diferencia del agrupamiento basado en densidades, los algoritmos anteriores no son capaces de obtener buenos resultados cuando se aplican a tareas con *clusters* de forma arbitraria o *clusters* dentro de *clusters.*

- Es capaz de detectar *outliers*, es decir, regiones de baja densidad cuyos puntos no serán introducidos en ningún *cluster*.

4. Agrupamiento basado en densidades - DBSCAN

- Trabaja con la idea de que si un punto pertenece a un *clúster*, este debería estar rodeado de un montón de otros puntos en ese *cluster*.
- Parámetros: El radio ϵ determina la longitud específica para mirar alrededor del punto seleccionado. Por otro lado, el número de puntos mínimos M que debe haber alrededor de otra observación para definir un *cluster*.



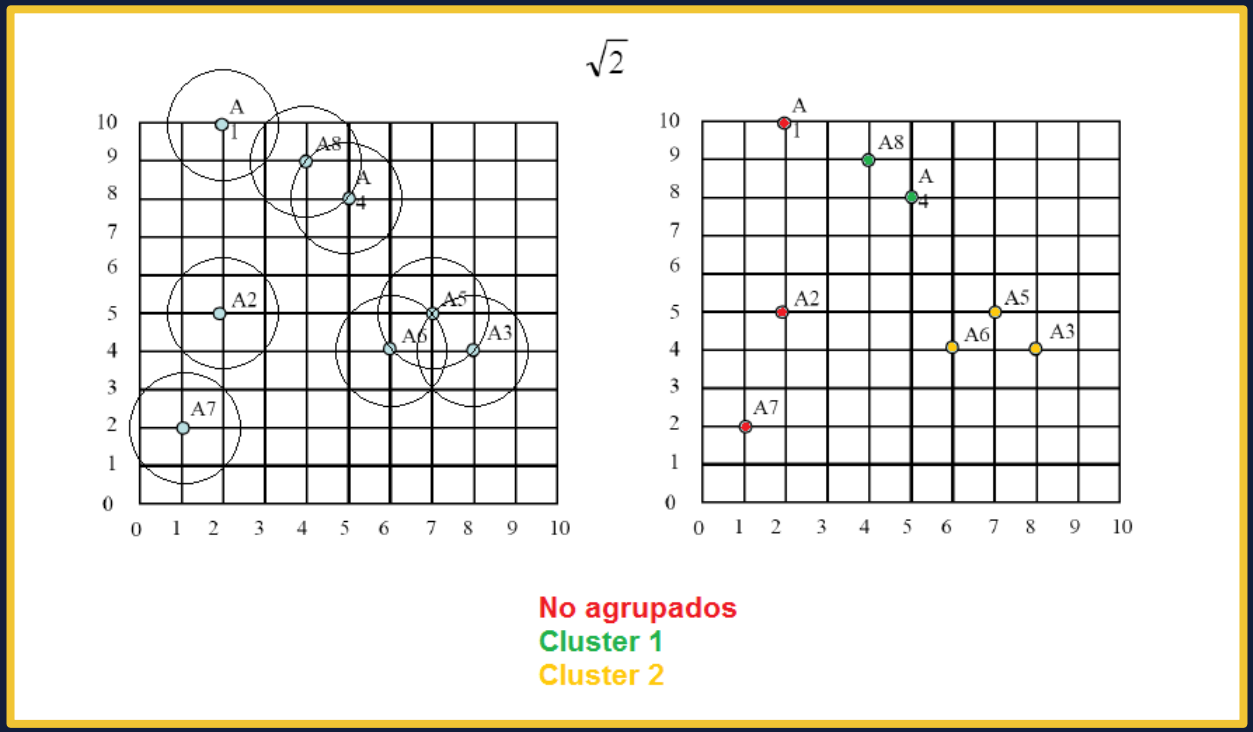
4. Agrupamiento basado en densidades – Ventajas e inconvenientes

- Identifica *clusters* de forma arbitraria.
- Son robustos a la presencia de ruido.
- No es necesario fijar el valor k, pero sí que se tiene que fijar ϵ y M.
- Es escalable, ya que sólo necesita un único recorrido sobre el conjunto de datos.
- Un problema es que los puntos fronterizos pertenezcan a más de un *cluster* y, por tanto, se tenga que decidir a que *cluster* pertenecen resultando un algoritmo no determinista.
- Si existen grandes diferencias en las densidades, DBSCAN puede que no trabaje bien ya que es complicado escoger un conjunto de parámetros que funcionen bien para todos los grupos.

4. Agrupamiento basado en densidades – Ejemplo

Aplicar DBSCAN sobre los puntos del ejemplo de la sección 2:

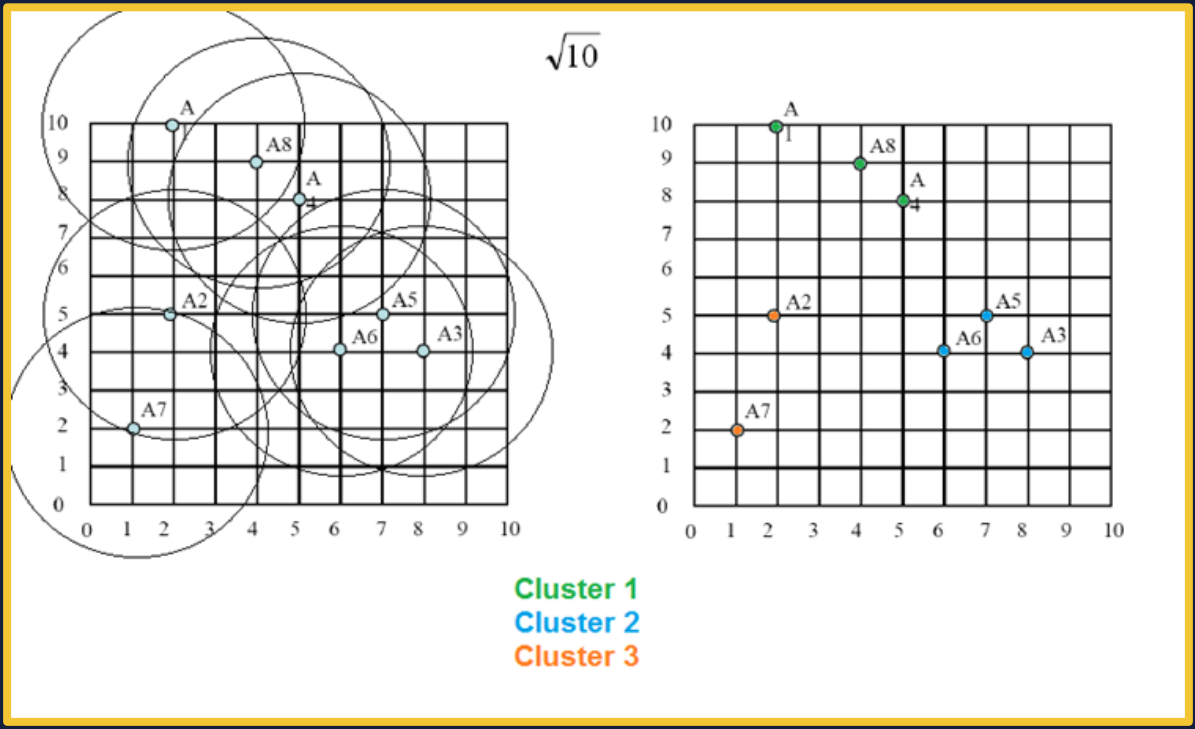
1. Considerando $M = 2$ y $\epsilon = \sqrt{2}$.



4. Agrupamiento basado en densidades – Ejemplo

Aplicar DBSCAN sobre los puntos del ejemplo de la sección 2:

1. Considerando $M = 2$ y $\epsilon = \sqrt{10}$.



MUCHAS GRACIAS POR SU ATENCIÓN

eip ESCUELA INTERNACIONAL POSGRADOS

aduran@grupomainjobs.com

Antonio Manuel Durán Rosal
https://www.linkedin.com/in/antonio-manuel-dur%C3%A1n-rosal-99ba92a5/

twitter.com/eiposgrados

facebook.com/eiposgrados

instagram.com/eiposgrados